Is Parameter Collision Hindering Continual Learning in LLMs?

Shuo Yang[1,*], **Kun-Peng Ning**[1,*], **Yu-Yang Liu**[1,†], **Jia-Yu Yao**[1],
Yong-Hong Tian[1,2], **Yibing Song**[3,4], **Li Yuan**[1,2,†]

[1]School of Electronic and Computer Engineering, Peking University
[2]Peng Cheng Laboratory   [3]DAMO Academy, Alibaba Group   [4]Hupan Lab
{shuo_yang, ningkp}@stu.pku.edu.cn,
{liuyuyang13, jiayu_yao, yhtian, yuanli-ece}@pku.edu.cn

Abstract

Large Language Models (LLMs) often suffer from catastrophic forgetting when learning multiple tasks sequentially, making continual learning (CL) essential for their dynamic deployment. Existing state-of-the-art (SOTA) methods, such as O-LoRA, typically focus on constructing orthogonality tasks to decouple parameter interdependence from various domains. In this paper, we reveal that building non-collision parameters is a more critical factor in addressing CL challenges. Our theoretical and experimental analyses demonstrate that non-collision parameters can provide better task orthogonality, which is a sufficient but unnecessary condition. Furthermore, knowledge from multiple domains will be preserved in non-collision parameter subspaces, making it more difficult to forget previously seen data. Leveraging this insight, we propose **N**on-collision **Lo**w-**R**ank **A**daptation (**N-LoRA**), a simple yet effective approach leveraging low collision rates to enhance CL in LLMs. Experimental results on multiple CL benchmarks indicate that N-LoRA achieves superior performance ($+\textbf{2.9}\%$), higher task orthogonality ($\times\textbf{4.1 times}$), and lower parameter collision ($\times\textbf{58.1 times}$) than SOTA methods.

1 Introduction

Continual Learning (CL) (De Lange et al., 2021; Silver et al., 2013) is essential for adapting large language models (LLMs) to new tasks, but it often leads to *catastrophic forgetting* (McCloskey and Cohen, 1989), where knowledge from previous tasks is overwritten. Various methods, such as rehearsal-based (Rebuffi et al., 2017), regularization-based (Kirkpatrick et al., 2017), and

architecture-based approaches (Mallya and Lazebnik, 2018), have been proposed to mitigate this, but they significantly increase computational demands due to the vast number of parameters in LLMs (Wu et al., 2024). Recent advances in parameter-efficient fine-tuning (PEFT), such as LoRA (Hu et al., 2021), introduce a minimal set of trainable parameters while maintaining the rest of the pre-trained LLM fixed. However, the task-specific parameters may collide with each other, potentially overwriting earlier knowledge and thus leading to poor model performance.

To tackle this challenge in LLMs, existing studies (Farajtabar et al., 2020; Saha et al., 2021; Wang et al., 2023a) attempted to employ orthogonal gradient descent-based methods to decouple parameter interdependence across different domains. Among them, O-LoRA (the current SOTA CL method in LLMs) incrementally updates task-specific parameters by constraining them to the orthogonal subspace of previous tasks, ensuring the independence of LoRA parameters across subspaces(Wang et al., 2023a). However, despite this independence, interactions between LoRA parameters in different subspaces may still occur, potentially leading to parameter collisions.

In this paper, we reveal that constructing non-collision LoRA parameters provides a more effective solution to the problem of parameter interdependence, as non-collision serves as a sufficient but not necessary condition for orthogonality. By avoiding unnecessary overlaps in the parameter space, non-collision inherently minimizes interference among task-specific parameters, ensuring better task isolation and knowledge retention. As illustrated in Figure 1(a), assuming a three-dimensional parameter space, existing methods (*e.g.* O-LoRA) aim to learn three mutually orthogonal (blue box) task vectors τ_1, τ_2, and τ_3. Each task vector $[x_i, y_i, z_i]$ occupies all three subspaces of the parameters, and these parameters will interact
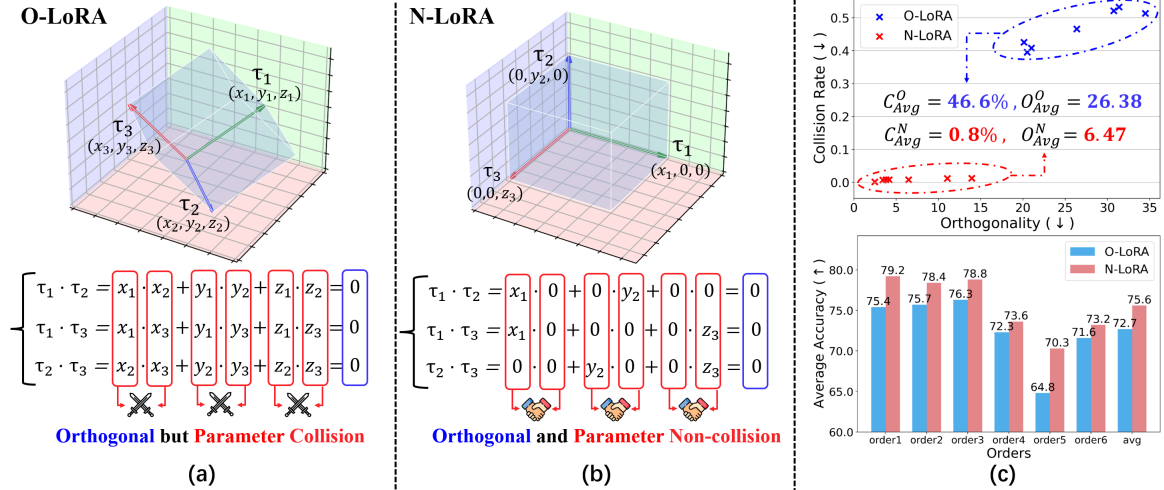

Figure 1: (a) Orthogonal but Parameter Collision: Tasks τ_1, τ_2, and τ_3 are mutually orthogonal but interaction within each space, resulting in parameter collision. (b) Non-collision and orthogonal: Tasks τ_1, τ_2, and τ_3 update only along distinct, non-conflicting subspaces, preserving prior task knowledge. (c) Performance Comparison: N-LoRA (red) and O-LoRA (blue) are compared across various metrics, with N-LoRA achieving lower collision rates, improved orthogonality, and superior average accuracy.

with each other (red box), hurting the model performance. In contrast, as illustrated in Figure 1(b), non-collision constructs an extremely sparse parameter space, allowing each task vector to be updated in a local subspace without affecting each other. Non-collision parameters naturally achieve better task orthogonality (blue box), effectively decoupling parameter interdependence while preserving prior knowledge.

Leveraging the above insights, we propose **N**on-collision **Lo**w-**R**ank **A**daptation (**N-LoRA**), a simple yet effective approach to tackling continual learning problems in LLMs. Instead of directly optimizing the collision rate, we introduce ℓ_1 constraints on task-specific LoRA parameters to make them extremely sparse, thereby significantly reducing the collision and dependency between various task vectors. This sparsity ensures that task-specific parameters operate independently in dedicated subspaces, avoiding interference and allowing for seamless task transitions. As illustrated in Figure 1(c), compared to the orthogonal-based method (O-LoRA), the proposed non-collision-based approach (N-LoRA) can significantly achieve better task orthogonality ($\times \mathbf{4.1}$ **times**), lower parameter collision ($\times \mathbf{58.1}$ **times**), and superior average accuracy ($+\mathbf{2.9\%}$) in all cases.

We empirically validate the effectiveness of the proposed N-LoRA through extensive experiments, demonstrating that it significantly outperforms prior SOTA methods in various datasets and

model architectures. The contribution of this paper can be summarized as follows:

- We reveal the intrinsic relationship between parameter orthogonality and collisions, clarifying that the key to addressing *catastrophic forgetting* lies in preventing parameter collisions, not just enforcing orthogonality.

- We introduce a simple yet effective method, called N-LoRA, to tackle the CL problem in LLMs, which leverages low collision parameters to enhance the model performance.

- Our experiments on various CL benchmarks demonstrate that N-LoRA not only surpasses previous SOTA methods, but also achieves better orthogonality and a lower collision rate.

- N-LoRA is compatible with existing PEFT-based approaches and serves as a plug-and-play solution that can substantially enhance the performance of existing SOTA methods.

2 Related Works

Parameter Efficient Fine-Tuning. Parameter Efficient Fine-Tuning (PEFT) (He et al., 2021; Ding et al., 2022) has become a vital area of research aimed at enhancing the performance of large language models (LLMs) by adjusting only a small portion of the model's parameters. The primary objective of PEFT is to achieve performance on

downstream tasks comparable to full model fine-tuning while minimizing computational resources by tuning only a small subset of model parameters. Various approaches have been proposed in this domain, including Adapter tuning (Houlsby et al., 2019), Prompt-based tuning (Wei et al., 2021), P-tuning (Liu et al., 2021a), Prefix tuning (Li and Liang, 2021), BitFit (Zaken et al., 2021), and LoRA (Wang et al., 2023a). In this work, we introduce an easy yet effective method built on LoRA to address this challenge, demonstrating strong performance.

Conventional Continual Learning. Conventional CL typically falls into three categories: (i) Rehearsal-based approaches (Li and Hoiem, 2017; Liu et al., 2021b; Lopez-Paz and Ranzato, 2017) store and retrain on previous task data, but can pose privacy concerns. (ii) Regularization-based approaches (Kirkpatrick et al., 2017; Zenke et al., 2017) mitigate forgetting by imposing constraints that discourage significant changes in key weights. (iii) Architecture-based approaches (Mallya and Lazebnik, 2018; Wang et al., 2023b; Liu et al., 2021c, 2023; Ning et al., 2024) expand model capacity dynamically or isolate existing weights with task-specific parameters to minimize interference between new and old tasks.

PEFT-Based Continual Learning for LLMs. LLMs with their vast number of parameters, create a significant computational burden for continual learning. To address this, methods like Learning to Prompt (L2P) (Wang et al., 2022b) use a pool of small, learnable prompts to guide pre-trained models through tasks, while DualPrompt (Wang et al., 2022a) combines expert and general prompts for task-specific and task-agnostic instructions. PEGP (Qiao et al., 2024) utilizes Singular Value Decomposition(SVD) of task data to control gradient updates through orthogonal gradient projection, ensuring that new parameters do not interfere with the established feature space. O-LoRA (Wang et al., 2023a) incrementally updates new task parameters by constraining them to the orthogonal subspace of previous tasks while fixing the LoRA parameters learned from past tasks to minimize catastrophic forgetting. However, although O-LoRA attempts to update parameters within an orthogonal subspace, it struggles to prevent parameter collisions, which ultimately fails to resolve the forgetting problem. Our proposed N-LoRA not only achieves better orthogonality than O-LoRA but also reduces parameter collisions and delivers superior performance.

3 Methodology

3.1 Continual Learning for LLMs

In this paper, we focus on continual learning for LLMs, where the model adapts sequentially to new tasks presented in a streaming manner without access to previous task data.

Orthogonality-Based CL for LLMs. Continual learning in LLMs poses computational challenges due to their large parameter space. To mitigate this, existing studies have used PEFT methods and employed orthogonal gradient descent-based approaches to decouple parameter interdependence across different tasks. Among these, O-LoRA introduces matrices $A \in \mathbb{R}^{d \times r}$ and, $B \in \mathbb{R}^{r \times k}$ based on the low-rank assumption (Wang et al., 2023a), updating the parameters as $W = W_0 + \Delta W = W_0 + AB$. O-LoRA enforces orthogonality between the task subspaces by imposing constraints on A_t and A_i, introducing an additional orthogonality loss:

$$L_{\text{orth}}(A_t) = \sum_{i=1}^{t-1} \left\| A_i^T A_t \right\|^2 \tag{1}$$

However, despite the enforced orthogonality constraints, the parameter subspaces used by different tasks may still overlap, preventing the achievement of full orthogonality (orthogonality loss fails to converge). Additionally, LoRA parameters within different subspaces may interact, leading to potential parameter collisions, which in turn degrade the performance across tasks.

3.2 Orthogonality vs. Non-Collision

We reveal that non-collision LoRA parameters provide a more effective solution to parameter interdependence and show that non-collision is a sufficient but not necessary condition for orthogonality (see Appendix A.1 for proof). The detailed analysis is as follows,

Definition 1. *For two parameter matrices ΔW_1 and ΔW_2 of the same dimensions, they are defined as non-collision if for every position (a, b), it holds that $\Delta W_1[a, b] = 0$ or $\Delta W_2[a, b] = 0$.*

Definition 2. *For two parameter matrices ΔW_1 and ΔW_2 of the same dimensions, they are said to be orthogonal if their matrix product satisfies $\Delta W_1^\top \Delta W_2 = 0$, where $\Delta W_1^\top$ denotes the transpose of ΔW_1.*

Theorem 1. *For two parameter matrices ΔW_1 and ΔW_2 of the same dimensions, non-collision is a*
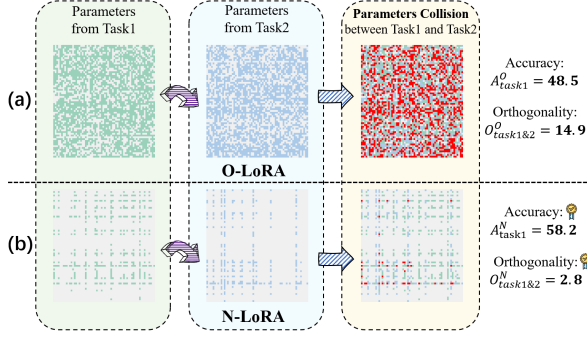
Figure 2: Parameter distribution of O-LoRA and N-LoRA. Green and blue represent parameters from Task 1 and Task 2, respectively; red indicates collisions. Lower orthogonality values signify better orthogonality. **(a) O-LoRA parameter distribution:** Despite the orthogonality constraint, significant parameter collisions occur, resulting in an accuracy of **48.5**% and an orthogonality of **14.9**. **(b) N-LoRA parameter distribution:** N-LoRA significantly reduces parameter collisions, achieving better performance with an accuracy of **58.2**% and an orthogonality of **2.8**.
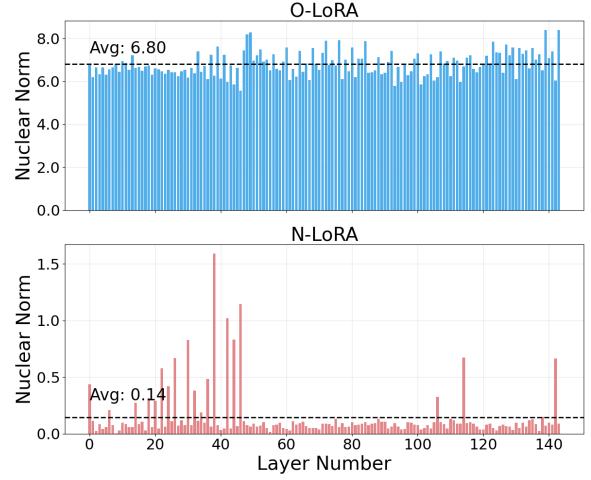

Figure 3: Nuclear norms of O-LoRA and N-LoRA. The top plot (blue) shows the nuclear norms for O-LoRA, indicating the subspace dimensionality used by each layer. The bottom plot (red) shows the nuclear norms for N-LoRA. The dashed lines in both plots represent the average nuclear norm across all layers.

sufficient but not necessary condition for orthogonality. Specifically,

$$\forall (a, b), \quad \Delta W_1[a, b] = 0 \vee \Delta W_2[a, b] = 0$$
$$\implies \Delta W_1^\top \Delta W_2 = 0,$$

Obviously, non-collision parameters inherently result in orthogonality. In other words, reducing parameter collisions is a more efficient way to achieve better task orthogonality.

Visualization on Parameter Collisions. Based on the above analysis, we aim to visualize the relationship between parameter orthogonality and non-collision. We conducted experiments on the T5-large model, comparing parameter distributions using O-LoRA (orthogonality-based) and N-LoRA (non-collision-based) in a continual learning setup for Task 1 and Task 2. In the visualizations, parameters near zero are shown in gray, non-zero parameters in green and blue, and collisions in red.

Figure 2 illustrates the parameter collisions under O-LoRA (top) and N-LoRA (bottom) methods. As shown in O-LoRA, despite the imposition of orthogonality constraints, significant parameter collisions (indicated by extensive red areas) were still observed, resulting in an accuracy of **48.5**% and an orthogonality measure of **14.9**. In contrast, N-LoRA effectively minimized parameter collisions (shown by small red areas), achieving an accuracy of **58.2**% and an orthogonality measure of **2.8**. Thus, N-LoRA's low collision rate not only leads

to higher accuracy but also results in better orthogonality.

3.3 Non-collision Low-Rank Adaptation

Building on the motivation above, the core goal of tackling continual learning in LLMs is to minimize parameter collisions as much as possible, *i.e.*,

$$\min \sum_{\forall (a,b)} \mathbf{1}\{\Delta W_1[a, b] \neq 0 \wedge ... \wedge \Delta W_n[a, b] \neq 0\},$$

where $\mathbf{1}\{\cdot\}$ is the indicator function. To achieve this, we propose to sparsify each LoRA parameter ΔW_i, *i.e.*,

$$\max \sum_{\forall (a,b)} \mathbf{1}\{W_i[a, b] = 0\} \tag{2}$$

We attempt to sparsify individual LoRA parameters to minimize overall collisions. We show that ***the collision rate will decrease quadratically with the sparsity rate*** in Appendix A.2. To further simplify, we introduce a sparsity constraint directly on ΔW_i while freezing the parameters from previous tasks. Specifically, for a sequence of tasks $T = \{t_1, t_2, \ldots, t_n\}$, a corresponding sequence of task-specific matrices $\Delta W = \{\Delta W_1, \Delta W_2, \ldots, \Delta W_n\}$ is obtained through continual fine-tuning using LoRA. For the i-th task, the matrix $\Delta W_i = A_i B_i$ is subject to ℓ_1 regularization:

$$L_{\text{sparse}} = \lambda \ell_1(\Delta W_i) = \lambda \|\Delta W_i\|_1 \tag{3}$$

where λ is the sparsity hyperparameter. For each new task, only the task-specific LoRA parameters are updated, while the pre-trained model and previously learned LoRA parameters remain fixed. Consequently, the overall loss function for the i-th task is defined as:

$$L = L_{\text{task}} + L_{\text{sparse}} = L_{t_i} + \lambda\|\Delta W_i\|_1 \quad (4)$$

Additionally, since the parameters from previous tasks are not needed during training or inference, they can be merged with the pre-trained model's parameters, effectively reducing memory usage and maintaining the model's efficiency.

Discussion: N-LoRA can effectively reduce the dimension of task subspaces. Beyond reducing collision rates, sparsity offers the additional advantage of effectively reducing the dimensionality of task subspaces. The nuclear norm, which provides an upper bound on the rank of a matrix (Li and Chen, 2021), corresponds to the number of subspaces utilized by each adapter in PEFT-based continual learning. Taking the first task of the Standard CL Benchmark as an example, we computed the nuclear norm of the LoRA weights inserted at each layer of the T5-large model for both O-LoRA and N-LoRA, as illustrated in Figure 3. With the rank set to 8 (i.e., 8 subspaces per task), O-LoRA (top plot) nearly fully occupied these subspaces, averaging **6.80** across all layers. In contrast, N-LoRA, by leveraging sparsity, utilized an average of only **0.14** subspaces. This clearly demonstrates that N-LoRA significantly reduces the dimension of task subspaces. Furthermore, knowledge from multiple domains will be preserved in these extremely low-dimensional (non-collision) parameter subspaces, making it more difficult to forget previously seen data.

4 Experiments

4.1 Experimental Setups

Datasets. To evaluate our method, we use the *Standard CL Benchmark* (Zhang et al., 2015) with five text classification datasets and the *Large Number of Tasks Benchmark* (Razdaibiedina et al., 2023) encompassing 15 datasets, including GLUE (Wang, 2018), SuperGLUE (Wang et al., 2019), and IMDB (Maas et al., 2011). We follow the experimental setups of LFPT5 (Qin and Joty, 2021) and O-LoRA (Wang et al., 2023a), using three different orders for each benchmark. Detailed task information

tion and configurations are in Appendix A.3, A.4, and A.6.

Metrics. Let $a_{i,j}$ be the test accuracy of the i-th task after training on the j-th task. A_i denotes the LoRA A matrix and $\Delta W_i = A_i B_i$ matrix, both after training on the i-th task. We evaluate the model using the following six metrics:

(I) **Average Accuracy (AA)** (Lopez-Paz and Ranzato, 2017): The average accuracy of all tasks after training on the final task, defined as:

$$A_T = \frac{1}{T}\sum_{i=1}^{T} a_{i,T} \quad (5)$$

(II) **Forgetting Rate (F.Ra)** (Chaudhry et al., 2018): This metric evaluates the amount of knowledge forgotten across the first $T-1$ tasks, as:

$$F_T = \frac{1}{T-1}\sum_{t=1}^{T-1}\left(\max_{k=t}^{T-1} a_{k,t} - a_{T,t}\right) \quad (6)$$

(III) **Orthogonal Overlap (OO)** (Wang et al., 2023a): Follow the O-LoRA, it quantifies the orthogonal overlap between the subspace of the T-th task and the subspaces of the previous $T-1$ tasks, calculated as:

$$O_T = \sum_{i=1}^{t-1}\left\|A_i^T A_t\right\|^2 \quad (7)$$

(IV) **Adapter Weight Orthogonality Magnitude (AWOM)** (Bhardwaj et al., 2024): This metric measures the degree of orthogonality between adapter weights, defined as:

$$AWOM_T = \sum_{i=1}^{T-1}\left\|\Delta W_T^\top \Delta W_i\right\|_2 \quad (8)$$

(V) **Generalized Sparsity Rate (GSR)** (Hurley and Rickard, 2009): This metric quantifies the sparsity of a matrix when none of its elements are zero. For detailed information and proof, see Appendix A.7. The formula is:

$$S_{q,p}^*(\Delta W_i) = (mn)^{1/p-1/q}\frac{\ell_q(\Delta W_i)}{\ell_p(\Delta W_i)}, (p > q > 0) \quad (9)$$

where $m \times n$ is the dimension of ΔW_i, and ℓ_p and ℓ_q are the respective norms. In our case, $p = 2$ and $q = 1$.

(VI) **Average Collision Rate (ACR)**: The concept of parameter collision was mentioned in (Kirkpatrick et al., 2017). Here, we formally define the

Benchmarks	Standard CL Benchmark				Large Number of Tasks			
Methods	Order-1	Order-2	Order-3	avg	Order-4	Order-5	Order-6	avg
ProgPrompt	75.2	75.0	75.1	75.1	78.0	77.7	77.9	77.9
PerTaskFT	70.0	70.0	70.0	70.0	78.1	78.1	78.1	78.1
MTL	80.0	80.0	80.0	80.0	76.5	76.5	76.5	76.5
SeqFT	18.9	24.9	41.7	28.5	7.4	7.4	7.5	7.4
SeqLoRA	44.6	32.7	53.7	43.7	0.6	1.9	1.6	1.6
IncLoRA	66.0	64.9	68.3	66.4	63.3	58.5	61.7	61.2
Replay	55.2	56.9	61.3	57.8	55.0	54.6	53.1	54.2
EWC	48.7	47.7	54.5	50.3	45.3	44.5	45.6	45.1
LwF	54.4	53.1	49.6	52.3	50.1	43.1	47.4	46.9
L2P	60.3	61.7	61.1	60.7	57.5	53.8	56.9	56.1
LFPT5	67.6	72.6	77.9	72.7	70.4	68.2	69.1	69.2
O-LoRA	75.4	75.7	76.3	75.8	72.3	64.8	71.6	69.6
N-LoRA	**79.2**	**78.4**	**78.8**	**78.8**	**73.6**	**70.3**	**73.2**	**72.4**

Table 1: Summary of results on two standard continual learning benchmarks using the T5-large model. The average accuracy after training on the final task is reported, with all results averaged over three independent runs.

evaluation metric ACR, which quantifies the Collision Rate (CR) between parameters across tasks. It is defined as:

$$ACR_T = \frac{2}{T(T-1)} \sum_{1 \leq i < j \leq T} CR_{i,j} \qquad (10)$$

where $CR_{i,j}$ is the collision rate between the i-th and j-th task' parameters, which is the fraction of elements that are non-zero at the same positions in both matrices, as defined in Definition 1:

$$CR_{i,j} = \frac{\sum_{a,b} \mathbf{1}\left(\Delta W_i[a,b] \neq 0 \land \Delta W_j[a,b] \neq 0\right)}{n \times m} \qquad (11)$$

Implementation Details Our experiments on the T5-large model were conducted on machines equipped with NVIDIA A6000 GPUs. For tasks in orders 1 to 3, we used a learning rate of 1e-3, a batch size of 32, a dropout rate of 0.1, and a weight decay rate of 0. The models were trained for 10 epochs with $\lambda = 0.4$. For tasks in order 4, we set the learning rate to 5e-4, the batch size to 32, the dropout rate to 0.1, and the weight decay rate to 0. The models were trained for 10 epochs, and λ values were set as follows: (0.4, 0.4, 1.2, 0.4, 0.4, 0.4, 1.2, 1.2, 1.2, 0.4, 1.2, 1.2, 0.4, 0.4, 1.2). For tasks in order 5, we used a learning rate of 8e-4, a batch size of 32, a dropout rate of 0.1, and a weight decay rate of 0. The models were trained for 10 epochs with $\lambda = 0.55$. For tasks in order 6, we set the learning rate to 1e-3, the batch size to 32,

the dropout rate to 0.1, and the weight decay rate to 0. The models were trained for 10 epochs with $\lambda = 1.8$.

Details for the specific task sequences in each order can be found in Appendix A.4. Details for experiments on the LLAMA-7B model can be found in Appendix A.6.

Baselines. To demonstrate the effectiveness of N-LoRA, we follow the setup of O-LoRA (Wang et al., 2023a) and compare it against the following baseline methods:

(i) Independent Training Methods: **PerTaskFT** trains a separate model for each task, while **ProgPrompt** (Razdaibiedina et al., 2023) learns independent prompts for each task, which essentially results in training a distinct model for each task.

(ii) Upper Bound: **MTL** represents multi-task learning, where a single model is trained on all tasks simultaneously. This method serves as the upper bound for continual learning.

(iii) Non-Continual Learning Methods: **SeqFT** (de Masson D'Autume et al., 2019) fine-tunes all model parameters across a sequence of tasks. **SeqLoRA** trains a single LoRA adapter across the task sequence while freezing the pretrained model parameters. **IncLoRA** trains a new LoRA module for each task.

(iv) Continual Learning Methods: **Replay** fine-tunes all model parameters while replaying samples from previous tasks using a memory buffer. **EWC** (Kirkpatrick et al., 2017) applies a regularization loss to adjust all model parameters. **LwF** (Li and Hoiem, 2017) preserves knowledge from previous tasks by recording their responses on new task data and using these responses as a regular-

---
[3]With the exception of the results for O-LoRA and N-LoRA, which were obtained as part of this work, all other data in Table 1 are based on those reported by (Wang et al., 2023a), using the same experimental setup.

Benchmarks	Order-1	Order-2	Order-3	avg
O-LoRA	76.8	75.7	75.7	76.1
N-LoRA	**77.2**	**77.3**	**78.4**	**77.6**

Table 2: Performance comparison between N-LoRA and O-LoRA on the larger LLaMA-7B model, reporting average accuracy across all task orders. Results are averaged over three independent runs.

ization term. Both **L2P** (Wang et al., 2022b) and **LFPT5** (Qin and Joty, 2021) employ prompts, dynamically selecting or generating prompts to adapt to new tasks. **O-LoRA** (Wang et al., 2023a) incrementally updates parameters for new tasks by constraining them to the orthogonal subspace of previous tasks while keeping the LoRA parameters learned from prior tasks fixed.

4.2 Comparisons with SOTA methods

Tables 1 and 2 present the performance comparison (Average Accuracy A_T) between N-LoRA and other baselines on different benchmarks. Following LFPT5 (Qin and Joty, 2021) and O-LoRA (Wang et al., 2023a), we report the results from three independent runs using different task orders and models on the CL benchmark.

Performance on Standard CL Benchmarks. N-LoRA consistently outperformed all previous methods across all task orders by a significant margin on standard CL benchmarks. In particular, N-LoRA achieved over a **3.0**% improvement compared to the previous SOTA method, O-LoRA. Additionally, N-LoRA surpassed independent training methods such as ProgPrompt and PerTaskFT, and its performance approached that of MTL, the upper bound for continual learning. This indicates that N-LoRA is capable of efficiently leveraging knowledge from multiple tasks. As shown in Table 2, N-LoRA also significantly outperformed O-LoRA on the larger LLaMA-7B model, demonstrating that our method remains effective even on larger-scale LLMs.

Performance on a Large Number of Tasks. Continual learning with a large number of tasks presents a significantly more challenging benchmark. Among the evaluated methods, SeqLoRA exhibits the worst performance due to significant parameter collisions and overlaps in non-orthogonal spaces. Its metrics, including OO (3796.23), AWOM (1120.5), Sparsity Rate (0.682), and ACR (0.542), reflect severe inefficiencies, limiting its ability to maintain task-specific knowledge.
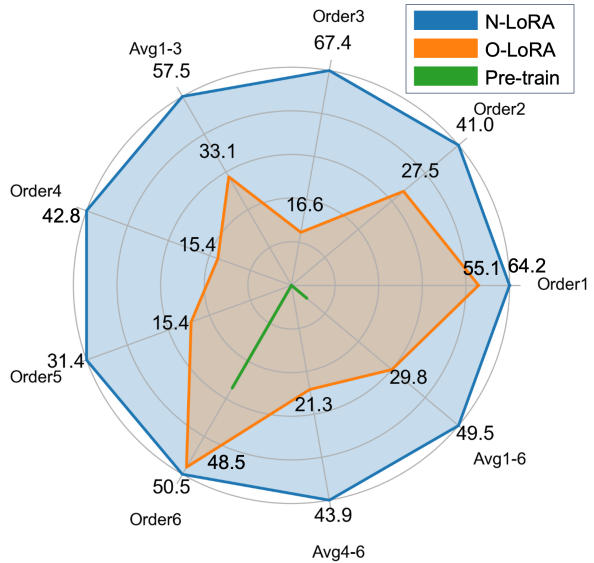

Figure 4: Performance comparison on unseen tasks on Average Accuracy metric. N-LoRA consistently outperforms O-LoRA across all task order settings. The pre-trained model performs poorly, with accuracy close to random on unseen tasks.

In contrast, as shown in Table 1, N-LoRA surpasses all existing SOTA methods across task orders (4, 5, and 6), achieving an average performance improvement of **2.8**%. By reducing parameter collisions and optimizing subspace allocation, N-LoRA delivers significantly better metrics, outperforming SeqLoRA by factors of approximately 1k, 8k, 85, and 71, respectively. These results underscore N-LoRA's superior capacity to handle numerous tasks while preserving task-specific knowledge. Further comparisons with O-LoRA are detailed in Appendix A.5.

4.3 Enhancing Generalization of LLMs

This subsection explores how N-LoRA affects the generalization ability of LLMs, evaluated on the Standard CL Benchmark and Large Number of Tasks Benchmark. Using both N-LoRA and O-LoRA (current SOTA method), we conduct continual learning across the first $T-1$ tasks, followed by testing on the unseen t-th task. For comparison, we also evaluate pre-trained LLMs directly on the unseen task.

As shown in Figure 4, the pre-trained models perform nearly at random, achieving close to zero accuracy on the unseen task, while N-LoRA achieves an average performance of **49.54**%. Moreover, N-LoRA delivers an average performance improvement of **+19.78**% over O-LoRA, consistently out-

Metrics	OO($\downarrow$)		AWOM($\downarrow$)	
Order	O-LoRA	N-LoRA	O-LoRA	N-LoRA
1	21.00	**3.80**	55.42	**0.13**
2	20.50	**4.16**	126.25	**0.17**
3	20.13	**3.42**	118.75	**0.15**
4	30.75	**13.94**	14.40	**0.19**
5	31.38	**11.06**	14.40	**0.19**
6	34.50	**2.47**	6.57	**0.03**
avg	26.38	**6.47**	55.96	**0.14**

Table 3: Performance comparison between O-LoRA and N-LoRA on Orthogonal Overlap (OO) and Adapter Weight Orthogonality Magnitude (AWOM) metrics. Lower values indicate better orthogonality.

performing it across all orders (1-6). These findings underscore the effectiveness of N-LoRA in enhancing strong generalization to unseen tasks.

4.4 N-LoRA vs O-LoRA: Reducing Forgetting

In this subsection, we conduct an in-depth analysis of how N-LoRA reduces forgetting from multiple perspectives, providing a comprehensive comparison with O-LoRA, the current SOTA method.

Higher Orthogonality. We adopt Orthogonal Overlap (OO) (Wang et al., 2023a) and Adapter Weight Orthogonality Magnitude (AWOM) (Bhardwaj et al., 2024) as metrics to assess orthogonality. As shown in Table 3, N-LoRA surpasses O-LoRA across both metrics, achieving $\times$**4.07 times** and $\times$**388.3 times** improvements, respectively. These results further demonstrate that the low collision rate induced by sparsity constraints can achieve stronger orthogonality compared to orthogonality constraints. Notably, although O-LoRA directly optimizes orthogonality using Orthogonal Overlap (OO) as its loss function, its actual OO is still lower than that of N-LoRA. This can likely be attributed to significant overlap in the task subspaces, leading to unavoidable conflicts, despite the orthogonal constraint. In contrast, as discussed in Section 3.3, N-LoRA reduces the dimensionality of task subspaces, decreasing the collison rate and consequently enhancing orthogonality.

Lower Sparsity Leads to Fewer Collisions and Reduced Forgetting. Figure 5 illustrates the relationship between sparsity, collision rate, and forgetting rate for N-LoRA and O-LoRA. It can be observed that the data points for N-LoRA are clustered in the lower-left corner of the plot, with an average forgetting rate of **2.37**%. In contrast, the data points for O-LoRA are in the upper-right corner, with an average forgetting rate of **4.20**%. This
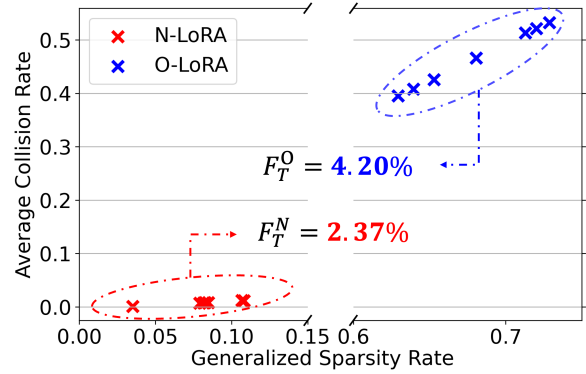

Figure 5: The relationship between sparsity, collision rate, and forgetting. The x-axis and y-axis represent the Generalized Sparsity and Average Collision Rate, respectively. Data points for N-LoRA (red) are clustered in the lower-left corner, while O-LoRA (blue) data points are concentrated in the upper-right corner. The dashed ellipses indicate the average forgetting rates for each method.

indicates that across varying benchmarks and orders, lower sparsity leads to fewer collisions and consequently reduces forgetting. The sparsity constraint in N-LoRA effectively minimizes collision rates, directly contributing to lower forgetting. Conversely, even though O-LoRA applies an orthogonality constraint on subspaces, it still encounters a high collision rate, resulting in greater forgetting.

Plug-and-Play Integration. Beyond its standalone effectiveness, N-LoRA can be seamlessly integrated into existing PEFT-based continual learning methods, acting as a versatile plug-and-play solution. As shown in Table 4, integrating N-LoRA into O-LoRA, resulting in "O-LoRA + N-LoRA", led to significant performance improvements across all task orders on the Standard CL Benchmark, with increases of **4.3**%, **4.1**%, and **3.2**% for Order-1, Order-2, and Order-3, respectively. These results not only surpass independent training methods but also approach the upper performance limit set by MTL, which is 80%. Additionally, "O-LoRA + N-LoRA" outperformed O-LoRA on all task orders in the Large Number of Tasks Benchmark, further demonstrating its robustness in managing a wide range of tasks and consistently enhancing continual learning performance.

4.5 Ablation Studies

In this subsection, we perform ablation studies to investigate the impact of sparsity constraints and the various ways of applying them. As illustrated in Figure 6, the model experiences severe forget-

Benchmarks	Standard CL Benchmark				Large Number of Tasks			
Methods	Order-1	Order-2	Order-3	avg	Order-4	Order-5	Order-6	avg
O-LoRA	75.4	75.7	76.3	75.8	72.3	64.8	71.6	69.6
O-LoRA + N-LoRA	**79.7**	**79.8**	**79.1**	**79.5**	**72.9**	**67.2**	**74.0**	**71.4**

Table 4: Performance comparison of O-LoRA and O-LoRA+N-LoRA on two benchmarks. **O-LoRA+N-LoRA** represents the integration of N-LoRA into O-LoRA, enhancing the performance of O-LoRA across all task orders.
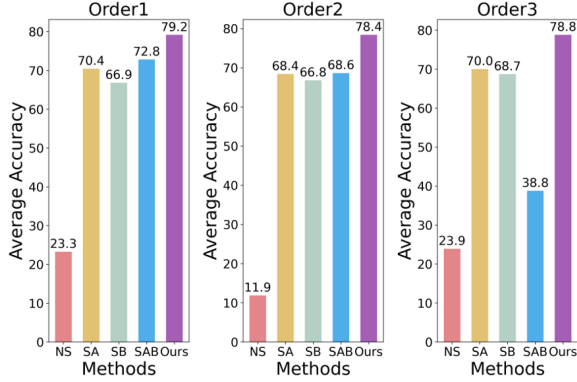

Figure 6: Ablation study comparing different sparsity constraint methods. NS refers to no sparsity constraint, while SA and SB apply sparsity constraints only to LoRA matrices A and B, respectively. SAB applies independent sparsity constraints to both matrices A and B. Ours represents the N-LoRA method, which significantly outperforms all other methods.

ting when no sparsity constraint is applied (NS method). Applying sparsity to only LoRA matrix A or matrix B (SA and SB methods) can partially enhance parameter updates, resulting in some performance gains. However, since ΔW is the product of both A and B, constraining only one matrix still permits considerable redundancy in the overall update. Simultaneously applying constraints to both A and B (SAB method) also fails to achieve optimal results. This is because the actual update during forward propagation is $\Delta W = AB$, and constraining the matrices independently hampers effective gradient flow, thereby impeding global optimization. The experimental results clearly show that only the N-LoRA approach (Ours) effectively reduces parameter collisions and substantially improves performance.

5   Conclusion

In this paper, we theoretically and empirically reveal that constructing non-collision parameters is a more critical factor in overcoming continual learning challenges. Building on this insight, we propose N-LoRA, a simple yet effective approach that leverages sparsity and low collision rates to improve continual learning. Extensive experiments on continual learning benchmarks demonstrate that, compared to SOTA methods, N-LoRA not only achieves better performance, higher task orthogonality, and lower parameter collision rates through sparsity, but also effectively reduces the dimensionality of task subspaces.

6   Limitations

Although N-LoRA demonstrates superior performance across multiple metrics in continual learning benchmarks, several limitations remain. In scenarios with a significantly larger number of tasks (e.g., hundreds or thousands), the parameter space can still become saturated, despite the imposed sparsity constraint. However, N-LoRA utilizes fewer subspaces than conventional LoRA, allowing it to reach saturation more slowly. Its scalability and effectiveness in such scenarios, as well as in highly dynamic real-world continual learning tasks, require further investigation.

The challenges of managing a large number of tasks and addressing concept drift remain open in the CL-LLM field. By minimizing parameter collisions and optimizing subspace allocation, N-LoRA demonstrates improved performance over longer sequences and more tasks. Effective subspace allocation also mitigates bias accumulation during concept drift, reducing abrupt performance drops and promoting more gradual declines. In future work, we will explore N-LoRA's scalability for larger tasks, longer sequences, and dynamic environments with concept drift.

Acknowledgments

References

Kartikeya Bhardwaj, Nilesh Prasad Pandey, Sweta Priyadarshi, Viswanath Ganapathy, Rafael Esteves, Shreya Kadambi, Shubhankar Borse, Paul Whatmough, Risheek Garrepalli, Mart Van Baalen, et al. 2024. Sparse high rank adapters. *arXiv preprint arXiv:2406.13175*.

Arslan Chaudhry, Puneet K Dokania, Thalaiyasingam Ajanthan, and Philip HS Torr. 2018. Riemannian walk for incremental learning: Understanding forgetting and intransigence. In *Proceedings of the European conference on computer vision (ECCV)*, pages 532–547.

Matthias De Lange, Rahaf Aljundi, Marc Masana, Sarah Parisot, Xu Jia, Aleš Leonardis, Gregory Slabaugh, and Tinne Tuytelaars. 2021. A continual learning survey: Defying forgetting in classification tasks. *IEEE transactions on pattern analysis and machine intelligence*, 44(7):3366–3385.

Cyprien de Masson D'Autume, Sebastian Ruder, Lingpeng Kong, and Dani Yogatama. 2019. Episodic memory in lifelong language learning. *Advances in Neural Information Processing Systems*, 32.

Ning Ding, Yujia Qin, Guang Yang, Fuchao Wei, Zonghan Yang, Yusheng Su, Shengding Hu, Yulin Chen, Chi-Min Chan, Weize Chen, et al. 2022. Delta tuning: A comprehensive study of parameter efficient methods for pre-trained language models. *arXiv preprint arXiv:2203.06904*.

Mehrdad Farajtabar, Navid Azizan, Alex Mott, and Ang Li. 2020. Orthogonal gradient descent for continual learning. In *International Conference on Artificial Intelligence and Statistics*, pages 3762–3773. PMLR.

Junxian He, Chunting Zhou, Xuezhe Ma, Taylor Berg-Kirkpatrick, and Graham Neubig. 2021. Towards a unified view of parameter-efficient transfer learning. *arXiv preprint arXiv:2110.04366*.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for nlp. In *International conference on machine learning*, pages 2790–2799. PMLR.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

Niall Hurley and Scott Rickard. 2009. Comparing measures of sparsity. *IEEE Transactions on Information Theory*, 55(10):4723–4741.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. 2017. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526.

Xiang Li and Songcan Chen. 2021. A concise yet effective model for non-aligned incomplete multi-view and missing multi-label learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(10):5918–5932.

Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. *arXiv preprint arXiv:2101.00190*.

Zhizhong Li and Derek Hoiem. 2017. Learning without forgetting. *IEEE transactions on pattern analysis and machine intelligence*, 40(12):2935–2947.

Xiao Liu, Kaixuan Ji, Yicheng Fu, Weng Lam Tam, Zhengxiao Du, Zhilin Yang, and Jie Tang. 2021a. P-tuning v2: Prompt tuning can be comparable to fine-tuning universally across scales and tasks. *arXiv preprint arXiv:2110.07602*.

Yaoyao Liu, Bernt Schiele, and Qianru Sun. 2021b. Adaptive aggregation networks for class-incremental learning. In *Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition*, pages 2544–2553.

Yuyang Liu, Yang Cong, Dipam Goswami, Xialei Liu, and Joost van de Weijer. 2023. Augmented box replay: Overcoming foreground shift for incremental object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11367–11377.

Yuyang Liu, Yang Cong, Gan Sun, Tao Zhang, Jiahua Dong, and Hongsen Liu. 2021c. L3doc: Lifelong 3d object classification. *IEEE Transactions on Image Processing*, 30:7486–7498.

David Lopez-Paz and Marc'Aurelio Ranzato. 2017. Gradient episodic memory for continual learning. *Advances in neural information processing systems*, 30.

Andrew Maas, Raymond E Daly, Peter T Pham, Dan Huang, Andrew Y Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies*, pages 142–150.

Arun Mallya and Svetlana Lazebnik. 2018. Packnet: Adding multiple tasks to a single network by iterative pruning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 7765–7773.

Michael McCloskey and Neal J Cohen. 1989. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of learning and motivation*, volume 24, pages 109–165. Elsevier.

Kun-Peng Ning, Hai-Jian Ke, Yu-Yang Liu, Jia-Yu Yao, Yong-Hong Tian, and Li Yuan. 2024. Sparse orthogonal parameters tuning for continual learning. *arXiv preprint arXiv:2411.02813*.

Jingyang Qiao, Zhizhong Zhang, Xin Tan, Yanyun Qu, Wensheng Zhang, and Yuan Xie. 2024. Gradient projection for parameter-efficient continual learning. *arXiv preprint arXiv:2405.13383*.

Chengwei Qin and Shafiq Joty. 2021. Lfpt5: A unified framework for lifelong few-shot language learning based on prompt tuning of t5. *arXiv preprint arXiv:2110.07298*.

Anastasia Razdaibiedina, Yuning Mao, Rui Hou, Madian Khabsa, Mike Lewis, and Amjad Almahairi. 2023. Progressive prompts: Continual learning for language models. *arXiv preprint arXiv:2301.12314*.

Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, Christoph H Lampert, et al. 2017. Incremental classifier and representation learning. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5533–5542.

Gobinda Saha, Isha Garg, and Kaushik Roy. 2021. Gradient projection memory for continual learning. *arXiv preprint arXiv:2103.09762*.

Daniel L Silver, Qiang Yang, and Lianghao Li. 2013. Lifelong machine learning systems: Beyond learning algorithms. In *2013 AAAI spring symposium series*.

Jianlin Su. 2023. How to measure the sparsity of data? Accessed: 2023-05-01.

Alex Wang. 2018. Glue: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2019. Superglue: A stickier benchmark for general-purpose language understanding systems. *Advances in neural information processing systems*, 32.

Xiao Wang, Tianze Chen, Qiming Ge, Han Xia, Rong Bao, Rui Zheng, Qi Zhang, Tao Gui, and Xuanjing Huang. 2023a. Orthogonal subspace learning for language model continual learning. *arXiv preprint arXiv:2310.14152*.

Zhicheng Wang, Yufang Liu, Tao Ji, Xiaoling Wang, Yuanbin Wu, Congcong Jiang, Ye Chao, Zhencong Han, Ling Wang, Xu Shao, et al. 2023b. Rehearsal-free continual language learning via efficient parameter isolation. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10933–10946.

Zifeng Wang, Zizhao Zhang, Sayna Ebrahimi, Ruoxi Sun, Han Zhang, Chen-Yu Lee, Xiaoqi Ren, Guolong Su, Vincent Perot, Jennifer Dy, et al. 2022a. Dual-prompt: Complementary prompting for rehearsal-free continual learning. In *European Conference on Computer Vision*, pages 631–648. Springer.

Zifeng Wang, Zizhao Zhang, Chen-Yu Lee, Han Zhang, Ruoxi Sun, Xiaoqi Ren, Guolong Su, Vincent Perot, Jennifer Dy, and Tomas Pfister. 2022b. Learning to prompt for continual learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 139–149.

Jason Wei, Maarten Bosma, Vincent Y Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. 2021. Finetuned language models are zero-shot learners. *arXiv preprint arXiv:2109.01652*.

Tongtong Wu, Linhao Luo, Yuan-Fang Li, Shirui Pan, Thuy-Trang Vu, and Gholamreza Haffari. 2024. Continual learning for large language models: A survey. *arXiv preprint arXiv:2402.01364*.

Elad Ben Zaken, Shauli Ravfogel, and Yoav Goldberg. 2021. Bitfit: Simple parameter-efficient fine-tuning for transformer-based masked language-models. *arXiv preprint arXiv:2106.10199*.

Friedemann Zenke, Ben Poole, and Surya Ganguli. 2017. Continual learning through synaptic intelligence. In *International conference on machine learning*, pages 3987–3995. PMLR.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. *Advances in neural information processing systems*, 28.

A Appendix

A.1 Proof of Theorem 1

In this section, we provide the detailed proof for the theorem presented in Section 3.2, which states that non-collision is a sufficient but not necessary condition for orthogonality.

Proof. **(1) Sufficiency:** Assume that ΔW_1 and ΔW_2 are non-collision. By definition, for any position (a, b), either $\Delta W_1[a, b] = 0$ or $\Delta W_2[a, b] = 0$. Consider any element (k, l) of the matrix product $\Delta W_1^\top \Delta W_2$:

$$(\Delta W_1^\top \Delta W_2)_{kl} = \sum_a \Delta W_1[a, k] \cdot \Delta W_2[a, l]$$

Since $\Delta W_1[a, k]$ and $\Delta W_2[a, l]$ cannot both be non-zero simultaneously, each term in the sum is zero. Thus, the entire sum is zero, implying:

$$\Delta W_1^\top \Delta W_2 = 0$$

Therefore, ΔW_1 and ΔW_2 are orthogonal.

(2) Non-necessity: Assume that ΔW_1 and ΔW_2 are orthogonal, i.e.,

$$\Delta W_1^\top \Delta W_2 = 0.$$

This implies that for any element (k, l) of the matrix product:

$$\sum_a \Delta W_1[a, k] \cdot \Delta W_2[a, l] = 0.$$

However, orthogonality only requires that the sum of these products equals zero. It does not imply that each individual product $\Delta W_1[a, k] \cdot \Delta W_2[a, l] = 0$ for all (a, k). As long as the sum is zero, some terms can be non-zero, leading to possible collisions. □

A.2 Relationship Between Collision Rate and Sparsity

Theorem 2. *Consider two $m \times n$ parameter matrices, ΔW_i and ΔW_j, with sparsity rates s_i and s_j, respectively. If the non-zero elements in each matrix are independently and randomly distributed, the collision rate $CR_{i,j}$ between these matrices is given by:*

$$CR_{i,j} = s_i \times s_j \qquad (12)$$

Thus, as the sparsity rate increases, the collision rate decreases quadratically.

Proof. **Assumption:** The positions of non-zero elements in matrices ΔW_i and ΔW_j are independent and randomly distributed.

At any position (a, b), the probability that both $\Delta W_i[a, b]$ and $\Delta W_j[a, b]$ are non-zero is $s_i \times s_j$. Since the elements are independently distributed, the expected total number of collisions in the matrix is the product of this collision probability and the total number of elements $(m \times n)$:

$$E_{\text{collisions}} = (m \times n) \times (s_i \times s_j) \qquad (13)$$

The collision rate $CR_{i,j}$ (as defined in section 4.1) is obtained by dividing the expected total number of collisions by the total number of elements:

$$CR_{i,j} = \frac{E_{\text{collisions}}}{m \times n} = \frac{(m \times n) \times s_i \times s_j}{m \times n} = s_i \times s_j \qquad (14)$$

Therefore, the collision rate $CR_{i,j}$ is directly proportional to the product of the sparsity rates s_i and s_j. Specifically, when $s_i = s_j = s$, the collision rate is:

$$CR_{i,j} = s^2 \qquad (15)$$

□

This confirms that the collision rate decreases as a quadratic function of the sparsity rate.

A.3 Datasets

In our experiments, we utilized 15 datasets, as detailed in Table 5, covering multiple evaluation metrics relevant to continual learning (CL) tasks. Specifically, the datasets were sourced from well-established benchmarks such as CL benchmark (Zhang et al., 2015), GLUE (Wang, 2018), and SuperGLUE (Wang et al., 2019). In addition, we incorporated the IMDB movie review dataset (Maas et al., 2011). These datasets span diverse tasks, including natural language inference (NLI), sentiment classification (SC), and topic classification (TC), ensuring a robust evaluation of the model's generalization across multiple tasks.

A.4 Task Sequence and Instructions

We report the task sequences used for CL experiments on the T5 and LLAMA models, as shown in Table 7. Additionally, Table 6 provides prompts for different tasks. NLI represents natural language inference tasks, including MNLI, RTE, and CB. SC refers to sentiment classification tasks, including Amazon, Yelp, SST-2, and IMDB, while TC denotes topic classification tasks, such as AG News, Dbpedia, and Yahoo.

No.	Dataset name	Category	Task	Domain	Metric
1	Yelp	CL Benchmark	sentiment analysis	Yelp reviews	accuracy
2	Amazon	CL Benchmark	sentiment analysis	Amazon reviews	accuracy
3	DBpedia	CL Benchmark	topic classification	Wikipedia	accuracy
4	Yahoo	CL Benchmark	topic classification	Yahoo Q&A	accuracy
5	AG News	CL Benchmark	topic classification	news	accuracy
6	MNLI	GLUE	NLI	various	accuracy
7	QQP	GLUE	paragraph detection	Quora	accuracy
8	RTE	GLUE	NLI	news, Wikipedia	accuracy
9	SST-2	GLUE	sentiment analysis	movie reviews	accuracy
10	WiC	SuperGLUE	word sense disambiguation	lexical databases	accuracy
11	CB	SuperGLUE	NLI	various	accuracy
12	COPA	SuperGLUE	QA	blogs, encyclopedia	accuracy
13	BoolQA	SuperGLUE	boolean QA	Wikipedia	accuracy
14	MultiRC	SuperGLUE	QA	various	accuracy
15	IMDB	SuperGLUE	sentiment analysis	movie reviews	accuracy

Table 5: Datasets and Their Tasks, Categories, Domains, and Metrics

Task	Prompts
NLI	What is the logical relationship between the "sentence 1" and the "sentence 2"? Choose one from the option.
QQP	Whether the "first sentence" and the "second sentence" have the same meaning? Choose one from the option.
SC	What is the sentiment of the following paragraph? Choose one from the option.
TC	What is the topic of the following paragraph? Choose one from the option.
BoolQA	According to the following passage, is the question true or false? Choose one from the option.
MultiRC	According to the following passage and question, is the candidate answer true or false? Choose one from the option.
WiC	Given a word and two sentences, whether the word is used with the same sense in both sentence? Choose one from the option.

Table 6: Task Descriptions and Corresponding Prompts

A.5 Detailed Comparison between N-LoRA and O-LoRA

To ensure a statistically significant comparison between O-LoRA and N-LoRA, we retrained and evaluated each experimental group five times as shown in Table 8. N-LoRA demonstrated superior performance over O-LoRA in terms of both mean and standard deviation. Additionally, N-LoRA demonstrates greater stability than O-LoRA, as evidenced by the smaller standard deviation across all task orders.

A.6 Implementation Details

For the experiments on the LLAMA-7B model, we used the following settings: For the tasks in order 1, we set the learning rate to (1e-3, 1e-4, 1e-4, 1e-4), the batch size to 4, the dropout rate to 0.1, and the weight decay rate to 0. The models were trained for (1, 10, 1, 1) epochs with $\lambda = 0.46$. For the tasks in order 2, the learning rate was set to (1e-4, 5e-5, 5e-5, 5e-5), the batch size to 4, the dropout rate to 0.1, and the weight decay rate to 0. The models were trained for (1, 5, 1, 1) epochs with $\lambda = 0.38$. For the tasks in order 3, we used a learning rate of 1e-4, a batch size of 4, a dropout rate of 0.1, and a weight decay rate of 0. The models were trained

Order	Model	Task Sequence
1	T5, LLaMA	dbpedia → amazon → yahoo → ag
2	T5, LLaMA	dbpedia → amazon → ag → yahoo
3	T5, LLaMA	yahoo → amazon → ag → dbpedia
4	T5	mnli → cb → wic → copa → qqp → boolqa → rte → imdb → yelp → amazon → sst-2 → dbpedia → ag → multirc → yahoo
5	T5	multirc → boolqa → wic → mnli → cb → copa → qqp → rte → imdb → sst-2 → dbpedia → ag → yelp → amazon → yahoo
6	T5	yelp → amazon → mnli → cb → copa → qqp → rte → imdb → sst-2 → dbpedia → ag → yahoo → multirc → boolqa → wic

Table 7: Task Sequences for Different Models

Benchmarks Methods	Standard CL Benchmark				Large Number of Tasks			
	Order-1	Order-2	Order-3	avg	Order-4	Order-5	Order-6	avg
O-LoRA	$75.03^{\pm 0.82}$	$76.12^{\pm 0.37}$	$76.58^{\pm 0.76}$	$75.91^{\pm 0.65}$	$72.39^{\pm 1.30}$	$64.55^{\pm 3.87}$	$71.03^{\pm 3.24}$	$69.32^{\pm 3.42}$
N-LoRA	$\mathbf{79.14^{\pm 0.41}}$	$\mathbf{78.29^{\pm 0.39}}$	$\mathbf{77.91^{\pm 1.93}}$	$\mathbf{78.45^{\pm 0.51}}$	$\mathbf{73.85^{\pm 0.53}}$	$\mathbf{70.09^{\pm 0.93}}$	$\mathbf{73.07^{\pm 1.42}}$	$\mathbf{72.34^{\pm 1.62}}$

Table 8: Detailed performance comparison between N-LoRA and O-LoRA on two standard continual learning benchmarks using the T5-large model. Results are reported as the mean and standard deviation (in the superscript) over five independent runs.

for (5, 10, 1, 1) epochs with $\lambda = 0.4$.

A.7 Supplementary Explanation For Generalized Sparsity Rate

Narrowly speaking, "sparsity" refers to the presence of a large number of zero elements in data, with the simplest metric being the proportion of zeros. However, in this paper, the parameter matrix ΔW is derived from the product of LoRA matrices B and A, making it unlikely for individual parameters to be exactly zero (as this would require the dot product of each row in B and each column in A to equal zero).

The generalized sparsity measure $S(x)$ should satisfy a series of ideal properties to ensure its theoretical validity and consistency with logical reasoning. Therefore, following the work of (Hurley and Rickard, 2009), we adopt a broader definition of sparsity, the sparsity measure $S(x)$ is considered to meet the following six key properties:

Property 1. _Robin Hood Property (D1)_ _When energy is transferred from a larger element to a smaller element in a vector, the sparsity should decrease. Assume in the vector x, $x_i > x_j$, and we transfer an amount α $(0 < \alpha < (x_i - x_j)/2)$ from x_i to x_j, resulting in a new vector x'. Then, the following should hold:_

$$S(x') < S(x)$$

This property reflects that a more evenly dis-

tributed vector is less sparse.

Property 2. _Dilation Invariance (D2)_ _For any positive scalar $\alpha > 0$, the sparsity measure should satisfy:_

$$S(\alpha x) = S(x)$$

This implies that sparsity is a relative property, and multiplying all elements of the vector by a positive constant does not change its sparsity.

Property 3. _Rising Tide Property (D3)_ _When a positive scalar α is added to all elements of the vector, the sparsity should decrease. Specifically:_

$$S(x + \alpha) > S(x), \ \alpha > 0$$

As all elements increase and approach non-zero values, the sparsity naturally decreases.

Property 4. _Cloning Invariance (D4)_ _If the vector x is replicated (cloned) one or more times, its sparsity should remain unchanged. For the cloned vector $x' = [x, x]$, the following should hold:_

$$S(x') = S(x)$$

This property reflects that duplicating elements in the vector does not affect its sparsity.

Property 5. _Dominance by Maximum Value (P1)_ _When one element x_i in the vector becomes sufficiently large, the sparsity should be dominated by that element. Specifically, as $x_i \to \infty$:_

$$S(x) \to max$$

This property indicates that when one element is overwhelmingly large, the sparsity of the vector is almost entirely determined by that element.

Property 6. *Effect of Adding Zeroes (P2) When zero elements are added to the vector, its sparsity should increase. If we add a zero to the vector, resulting in $x' = [x, 0]$, then:*

$$S(x') > S(x)$$

Adding zero elements increases the sparsity of the vector.

For a matrix ΔW_i of dimensions $m \times n$, we define the Generalized Sparsity Rate (GSR) as:

$$S^*_{q,p}(\Delta W_i) = (mn)^{1/p - 1/q} \frac{\ell_q(\Delta W_i)}{\ell_p(\Delta W_i)}, \quad (p > q > 0) \tag{16}$$

where ℓ_p and ℓ_q are the respective matrix norms, and $m \times n$ is the dimension of the matrix.

In this paper, we specifically use $p = 2$ and $q = 1$, which gives the following sparsity measure(Su, 2023):

$$S^*_{1,2}(\Delta W_i) = \frac{\sqrt{mn} \cdot \ell_1(\Delta W_i)}{\ell_2(\Delta W_i)} \tag{17}$$

where:

- $\ell_1(\Delta W_i) = \sum_{a=1}^{m} \sum_{b=1}^{n} |\Delta W_i[a,b]|$ denotes the ℓ_1 norm, which is the sum of the absolute values of all elements in the matrix.

- $\ell_2(\Delta W_i) = \left(\sum_{i=a}^{m} \sum_{j=b}^{n} |\Delta W_i[a,b]|^2 \right)^{1/2}$ denotes the ℓ_2 (Frobenius) norm of the matrix.

The sparsity measure $S^*_{1,2}(\Delta W_i)$ satisfies the six properties outlined above, ensuring its theoretical soundness and consistency. Specifically, it adheres to the following:

Theorem 3. *The sparsity measure $S^*_{1,2}(\Delta W_i)$ defined for a matrix satisfies the following six properties:*

1. Robin Hood Property (D1)
2. Dilation Invariance (D2)
3. Rising Tide Property (D3)
4. Cloning Invariance (D4)
5. Dominance by Maximum Value (P1)
6. Effect of Adding Zeroes (P2)

Next, we will demonstrate that the proposed formula for Generalized Sparsity Ratio (GSR) satisfies the properties outlined above.

A.7.1 Proof for Robin Hood Property (D1)

Statement: Transferring a value from a larger element to a smaller element in the matrix reduces the overall sparsity (i.e., $S^*_{1,2}(\Delta W'_i) > S^*_{1,2}(\Delta W_i)$).

Proof. Let ΔW_i be an $m \times n$ matrix, and suppose $\Delta W_i[a_1, b_1] > \Delta W_i[a_2, b_2]$. Transfer an amount α (where $0 < \alpha < \frac{\Delta W_i[a_1,b_1] - \Delta W_i[a_2,b_2]}{2}$) from $\Delta W_i[a_1, b_1]$ to $\Delta W_i[a_2, b_2]$, resulting in a new matrix $\Delta W'_i$:

$$\Delta W'_i[a_1, b_1] = \Delta W_i[a_1, b_1] - \alpha, \tag{18}$$
$$\Delta W'_i[a_2, b_2] = \Delta W_i[a_2, b_2] + \alpha, \tag{19}$$
$$\Delta W'_i[a, b] = \Delta W_i[a, b], \text{for all other } (a, b). \tag{20}$$

Compute $||\Delta W'_i||_1$: Since the total sum remains unchanged:

$$||\Delta W'_i||_1 = ||\Delta W_i||_1. \tag{21}$$

Compute $||\Delta W'_i||_2^2$:

$$\begin{aligned} ||\Delta W'_i||_2^2 &= ||\Delta W_i||_2^2 \tag{22}\\ &- (\Delta W_i[a_1,b_1]^2 - \Delta W_i[a_2,b_2]^2) \tag{23}\\ &+ (\Delta W'_i[a_1,b_1]^2 - \Delta W'_i[a_2,b_2]^2) \tag{24}\\ &= ||\Delta W_i||_2^2 - 2\alpha(\Delta W_i[a_1,b_1] - \Delta W_i[a_2,b_2]) + 2\alpha^2. \tag{25} \end{aligned}$$

Since $\Delta W_i[a_1, b_1] > \Delta W_i[a_2, b_2]$, the term $-2\alpha(\Delta W_i[a_1,b_1] - \Delta W_i[a_2,b_2])$ is negative, leading to:

$$||\Delta W'_i||_2^2 < ||\Delta W_i||_2^2 \implies ||\Delta W'_i||_2 < ||\Delta W_i||_2. \tag{26}$$

Compute $S^*_{1,2}(\Delta W'_i)$:

$$\begin{aligned} S^*_{1,2}(\Delta W'_i) &= \frac{\sqrt{mn} \cdot ||\Delta W'_i||_1}{||\Delta W'_i||_2} \tag{27}\\ &= \frac{\sqrt{mn} \cdot ||\Delta W_i||_1}{||\Delta W'_i||_2} \tag{28}\\ &> \frac{\sqrt{mn} \cdot ||\Delta W_i||_1}{||\Delta W_i||_2} = S^*_{1,2}(\Delta W_i). \tag{29} \end{aligned}$$

Therefore, $S^*_{1,2}(\Delta W'_i) > S^*_{1,2}(\Delta W_i)$, satisfying the Robin Hood Property.

$\square$

A.7.2 Proof for Dilation Invariance (D2)

Statement: For any positive scalar $\beta > 0$:

$$S_{1,2}^*(\beta \Delta W_i) = S_{1,2}^*(\Delta W_i). \qquad (30)$$

Proof. Consider the matrix after scaling, $\alpha \Delta W_i$. The ℓ_1 and ℓ_2 norms are given by:

$$\ell_1(\alpha \Delta W_i) = \sum_{a=1}^{m} \sum_{b=1}^{n} |\alpha \Delta W_i[a,b]| = \alpha \ell_1(\Delta W_i)$$

$$\ell_2(\alpha \Delta W_i) = \left(\sum_{a=1}^{m} \sum_{b=1}^{n} |\alpha \Delta W_i[a,b]|^2 \right)^{1/2}$$
$$= \alpha \ell_2(\Delta W_i)$$

Substituting these into the sparsity measure formula:

$$S_{1,2}^*(\alpha \Delta W_i) = \frac{\sqrt{mn} \cdot \ell_1(\alpha \Delta W_i)}{\ell_2(\alpha \Delta W_i)}$$
$$= \frac{\sqrt{mn} \cdot \alpha \ell_1(\Delta W_i)}{\alpha \ell_2(\Delta W_i)}$$
$$= \frac{\sqrt{mn} \cdot \ell_1(\Delta W_i)}{\ell_2(\Delta W_i)} = S_{1,2}^*(\Delta W_i)$$

Conclusion: Scaling the matrix does not affect the sparsity measure, which satisfies the dilation invariance property (D2).

$\square$

A.7.3 Proof for Rising Tide Property (D3)

Statement: Adding a positive scalar γ to all elements decreases sparsity:

$$S_{1,2}^*(\Delta W_i + \gamma) > S_{1,2}^*(\Delta W_i). \qquad (31)$$

Proof. Let $\Delta W_i' = \Delta W_i + \gamma$.
   Compute norms:

$$\|\Delta W_i'\|_1 = \|\Delta W_i\|_1 + mn\gamma, \qquad (32)$$
$$\|\Delta W_i'\|_2^2 = \|\Delta W_i\|_2^2 + 2\gamma\|\Delta W_i\|_1 + mn\gamma^2. \qquad (33)$$

Compute sparsity measure:

$$S_{1,2}^*(\Delta W_i') = \frac{\sqrt{mn} \cdot (\|\Delta W_i\|_1 + mn\gamma)}{\sqrt{\|\Delta W_i\|_2^2 + 2\gamma\|\Delta W_i\|_1 + mn\gamma^2}}. \qquad (34)$$

Since the numerator increases linearly and the denominator increases sub-linearly, $S_{1,2}^*(\Delta W_i') > S_{1,2}^*(\Delta W_i)$, satisfying the Rising Tide Property.

$\square$

A.7.4 Proof for Cloning Invariance(D4)

Statement: Replicating the matrix should not change its sparsity measure.

Proof. Assume that the matrix ΔW_i is replicated k times to obtain the matrix $\Delta W_i'$:

$$\Delta W_i' = [\Delta W_i \mid \Delta W_i \mid \cdots \mid \Delta W_i] \quad (k \text{ times})$$

The new matrix has dimensions $m \times (kn)$. The new ℓ_1 norm is:

$$\ell_1(\Delta W_i') = \sum_{a=1}^{m} \sum_{b=1}^{kn} |\Delta W_i'[a,b]|$$
$$= k \sum_{a=1}^{m} \sum_{b=1}^{n} |\Delta W_i[a,b]|$$
$$= k \ell_1(\Delta W_i)$$

The new ℓ_2 norm is:

$$\ell_2(\Delta W_i') = \left(\sum_{a=1}^{m} \sum_{b=1}^{kn} |\Delta W_i'[a,b]|^2 \right)^{1/2}$$
$$= \left(k \sum_{a=1}^{m} \sum_{b=1}^{n} |\Delta W_i[a,b]|^2 \right)^{1/2}$$
$$= \sqrt{k} \ell_2(\Delta W_i)$$

Substituting into the sparsity measure formula:

$$S_{1,2}^*(\Delta W_i') = \frac{\sqrt{mkn} \cdot \ell_1(\Delta W_i')}{\ell_2(\Delta W_i')}$$
$$= \frac{\sqrt{mkn} \cdot k\ell_1(\Delta W_i)}{\sqrt{k}\ell_2(\Delta W_i)}$$
$$= \frac{\sqrt{mn} \cdot \ell_1(\Delta W_i)}{\ell_2(\Delta W_i)}$$
$$= S_{1,2}^*(\Delta W_i)$$

Conclusion: Cloning the matrix does not affect the sparsity measure, satisfying the cloning invariance property (D4).

$\square$

A.7.5 Proof for Dominance by Maximum Value (P1)

Statement: When one element becomes very large, sparsity is dominated by that element.

Proof. Assume that one element $\Delta W_i[j,k]$ in the matrix becomes infinitely large, while all other elements remain unchanged. Then:

$$\ell_1(\Delta W_i) \approx |\Delta W_i[j,k]| + \sum_{(a,b) \neq (j,k)} |\Delta W_i[a,b]|$$

$$\ell_2(\Delta W_i) \approx \sqrt{|\Delta W_i[j,k]|^2 + \sum_{(a,b) \neq (j,k)} |\Delta W_i[a,b]|^2}$$
$$\approx |\Delta W_i[j,k]|$$

Thus, the sparsity measure becomes:

$$S_{1,2}^*(\Delta W_i) \approx \frac{\sqrt{mn} \cdot |\Delta W_i[j,k]|}{|\Delta W_i[j,k]|} = \sqrt{mn}$$

As $\Delta W_i[j,k] \to \infty$, the sparsity measure approaches its maximum value $\sqrt{mn}$.

Conclusion: The sparsity measure is dominated by the largest element, satisfying the Dominance by Maximum Value property (P1).

$\square$

A.7.6 Proof for Effect of Adding Zeroes (P2)

Statement: Adding zeroes increases sparsity.

Proof. Assume we add an all-zero column to the matrix ΔW_i, resulting in a new matrix $\Delta W_i'$ with dimensions $m \times (n+1)$.

The new ℓ_1 norm is:

$$\ell_1(\Delta W_i') = \ell_1(\Delta W_i) + \sum_{a=1}^{m} |0| = \ell_1(\Delta W_i)$$

The new ℓ_2 norm is:

$$\ell_2(\Delta W_i') = \left(\sum_{a=1}^{m} \sum_{b=1}^{n+1} |\Delta W_i'[a,b]|^2 \right)^{1/2}$$
$$= \left(l_2(\Delta W_i)^2 + \sum_{a=1}^{m} |0|^2 \right)^{1/2}$$
$$= \ell_2(\Delta W_i)$$

However, since the dimensions of the matrix have changed from $m \times n$ to $m \times (n+1)$, the sparsity measure formula becomes:

$$S_{1,2}^*(\Delta W_i') = \frac{\sqrt{m(n+1)} \cdot \ell_1(\Delta W_i')}{\ell_2(\Delta W_i')}$$
$$= \frac{\sqrt{m(n+1)} \cdot \ell_1(\Delta W_i)}{\ell_2(\Delta W_i)}$$
$$> \frac{\sqrt{mn} \cdot \ell_1(\Delta W_i)}{\ell_2(\Delta W_i)}$$
$$= S_{1,2}^*(\Delta W_i)$$

Since $\sqrt{m(n+1)} > \sqrt{mn}$, we conclude:

$$S_{1,2}^*(\Delta W_i') > S_{1,2}^*(\Delta W_i)$$

Conclusion: Adding zero elements increases the sparsity measure, satisfying the Effect of Adding Zeroes property (P2).

$\square$